

**AMENDMENTS TO THE CLAIMS:**

Kindly amend claim 11, as shown below.

This listing of claims will replace all prior versions and listings of claims in the Application:

**Claim 1 (previously presented):** A system for extending interactivity of a presentation markup language, the system comprising:

a collection of designated elements defined using a markup language, the collection of designated elements comprising one or more flow control elements for controlling statement flow of a web application, each designated element comprising:

a namespace; and

attributes for describing features of the designated element; and

a collection of associated instructions for performing functions to elements in thea document object model, the instructions associated with the designated elements.

**Claim 2 (previously presented):** The system as claimed in claim 1, further comprising an initialization function for directing the processing of one or more designated elements in the document object model, having instructions for traversing each node in the document object model, identifying designated elements having names following a predetermined naming convention, and calling functions associated with the identified designated elements.

**Claim 3 (cancelled).**

**Claim 4 (original):** The system as claimed in claim 1, further comprising:

a collection of designated attributes applied to one or more of the document object model elements for applying passive behavior to objects in the web application; and

a collection of associated instructions for performing functions associated with the designated attributes.

**Claim 5 (original)** The system as claimed in claim 4, wherein the collection of designated attributes comprises one or more of:

- a 'drag' attribute for specifying whether the element is movable by clicking and dragging it with the mouse;
- a 'pan' attribute for specifying whether the element is immune to panning;
- a 'zoom' attribute for specifying whether the element is immune to zooming;
- a 'selected' attribute for specifying whether the element has been selected; and
- a 'selectionGroup' attribute for specifying an 'id' attribute of a <selection> element that this element is associated with.

**Claim 6 (previously presented):** The system as claimed in claim 1, wherein the collection of designated elements further comprises one or more of:

- coordinate mapping elements for manipulating coordinates of objects in the web application;
- behavior elements for manipulating viewer behavior with respect to the web application;
- a focus element for selecting a group of elements in the web application; and
- a constraint element for constraining manipulable attributes of an element in a web application.

**Claim 7 (previously presented):** The system as claimed in claim 1, wherein the flow control elements comprise one or more of:

- an if element for defining a simple conditional statement which, when it evaluates to true, results in its child elements being executed or rendered;

a switch element for defining a conditional statement, and for comparing one value to other values defined in child <case> elements;

a case element for defining the value to compare to a 'value' attribute of the switch element;

a default element for containing action elements to be executed; and

a loop element for defining a repeated sequence of actions.

**Claim 8 (original):** The system as claimed in claim 6, wherein the coordinate mapping elements comprise one or more of:

a mousePosition element for defining a container for holding current mouse coordinates;

a mapCoords element for defining an object used for mapping coordinates in one space to another space, via a polynomial transformation, whose coefficients are determined by the coordinates of point-pairs;

a pointPair element for defining x-y coordinates for a same location in two different coordinate spaces; and

a mapProj element for defining an object used for mapping coordinates in one projection system to another.

**Claim 9 (original):** The system as claimed in claim 6, wherein the viewer behavior elements comprise one or more of:

a zoom element for scaling a document by a factor;

a pan element for translating a document by an amount; and

a playSound element for playing an audio file.

**Claim 10 (previously presented):** A method of extending interactivity of presentation markup languages, the method comprising

controlling statement flow of a web application, the method comprising the steps of:  
searching for a flow control element in a document object model of the web application;  
generating a function name associated with the flow control element;  
calling the generated function name; and  
processing child elements of the flow control element.

**Claim 11 (currently amended):** A method of extending interactivity of a presentation markup language, the method comprising the steps of:

searching for a designated control element, having a name which follows a designated naming convention, in a document object model; and  
calling a function, having a function name based on the naming convention, associated with the designated control element.

**Claim 12 (original):** The method as claimed in claim 11, wherein the step of searching includes the steps of:

traversing each node in the document object model; and  
determining whether an element has a name which follows a designated naming convention.

**Claim 13 (original):** The method as claimed in claim 11, wherein the step of calling a function includes the steps of:

dynamically generating a function name associated with the designated element;  
passing an object associated with the designated element as a parameter of the generated function;  
retrieving the attributes of the object; and  
performing a function stored in memory having the generated function name.

**Claim 14 (original):** The method as claimed in claim 13, wherein the step of dynamically generating includes the steps of:

determining if the name of the designated element contains a designated prefix;  
generating a function name comprising of the name of the designated element;  
assigning an object associated with the designated element as the parameter of the function; and  
assigning predetermined instructions of the designated element as steps for the function to perform.

**Claim 15 (original):** The method as claimed in claim 11, wherein the step of calling a function includes the steps of:

determining which script in a collection of scripts is associated with the designated element; and  
calling the script.

**Claim 16 (original):** The method as claimed in claim 11, further comprising the steps of:  
searching for a designated attribute in an element in a document object model; and  
calling a script associated with the designated attribute.

**Claim 17 (original):** The method as claimed in claim 16, wherein the step of searching for a designated attribute comprises the steps of:

searching attributes of an element in a document object model;  
determining whether an element attribute has a name which follows a designated naming convention.

**Claim 18 (original):** The method as claimed in claim 16, wherein the step of calling a script includes the steps of:

determining if the name of the designated attribute contains a designated prefix;

generating a function name comprising of the name of the designated attribute;  
assigning an object associated with the designated attribute as the parameter of the  
function name; and  
assigning predetermined instructions of the designated attribute as steps for a function  
having the function name to perform.

**Claim 19 (original):** The method as claimed in claim 16, wherein the step of calling a script  
includes the steps of:

dynamically generating a function name associated with the designated attribute;  
passing an object associated with the designated attribute as a parameter of the  
generated function name;  
receiving the attributes of the object; and  
performing a function stored in memory having the generated function name.

**Claim 20 (original):** The method as claimed in claim 19, wherein the step of dynamically  
generating comprises the steps of:

determining if the name of the designated attribute contains a designated prefix;  
generating a function name comprising of the name of the designated attribute;  
assigning an object associated with the designated attribute as the parameter of the  
function; and  
assigning predetermined instructions of the designated attribute as steps for the function  
to perform.

**Claim 21 (original):** The method as claimed in claim 19, wherein the step of calling a script  
includes the steps of:

determining which script in a collection of scripts is associated with the designated  
attribute; and

calling the script.

**Claim 22 (cancelled).**

**Claim 23 (previously presented):** The method as claimed in claim 10, further comprising one or more of the following:

coordinate mapping of a web application, the method comprising the steps of:

searching for a coordinate mapping element in a document object model of the web application;

generating a function name associated with the coordinate mapping element; and

calling the generated function name;

manipulating viewer behavior with respect to a web application, the method comprising the steps of:

searching for a viewer behavior element in a document object model of the web application;

generating a function name associated with the viewer behavior element; and

calling the generated function name;

focusing a group of elements in a web application, the method comprising the steps of:

searching for a focus element in a document object model of the web application;

generating a function name associated with the focus element; and

calling the generated function name;

constraining manipulable attributes of an element in a web application, the method comprising the steps of:

searching for a constraint element in a document object model of the web

application;

generating a function name associated with the constraint element; and

calling the generated function name; and

applying passive behavior to an element of a web application, the method comprising the steps of:

searching for a designated attribute of the element in a document object model of the web application;

generating a function name associated with the designated attribute; and

calling the generated function name.